abmR: an R package for agent-based model analysis of large-scale movements across taxa

Benjamin Gochanour[1,2,*], Javier Fernández-López[3], Andrea Contina[4]

[1] Corix Plains Institute, University of Oklahoma, Five Partners Place, 201 Stephenson Parkway, Norman, OK 73019, USA

[2] Oklahoma Biological Survey, University of Oklahoma, 111 East Chesapeake Street, SC Building 134, Norman, OK 73019, USA.

[3] University of Massachusetts Boston, Department of Biology, Integrated Sciences Complex, 100 Morrissey Blvd., Boston, MA 02125

[4] University of Colorado Denver, Department of Integrative Biology, Science Building 2074, Denver, CO 80217, USA

*Corresponding author

[*]E-mail: gocha001@umn.edu

Benjamin Gochanour ORCID: 0000-0002-0950-1739

Andrea Contina ORCID: 0000-0002-0484-6711

Abstract

1. Agent-based modeling (ABM) shows promise for animal movement studies. However, a robust, open-source, and spatially explicit ABM coding platform is currently lacking.
2. We present abmR, an R package for conducting continental-scale ABM simulations across animal taxa. The package features two movement functions, each of which relies on the Ornstein-Uhlenbeck (OU) model.
3. The theoretical background for abmR is discussed and the main functionalities are illustrated using two example populations.
4. Potential future additions to this open-source package may include the ability to specify multiple environmental variables or to model interactions between agents. Additionally, updates may offer opportunities for disease ecology and integration with other R movement modeling packages.

Keywords:

Animal migration; R programming; computer simulations; ecology

Abbreviated Title: agent-based models in R

1. INTRODUCTION

Animal movement is a complex behavioral trait that affects the survival of populations and species across taxa (Berg, 1983; Dingle, 2014). Long- and short-distance movements can be predictable, allowing populations to take advantage of seasonal food resources (e.g., migration), or more opportunistic, such as in the case of dispersal behaviors aimed at avoiding predators or finding potential mates (Giuggioli & Bartumeus, 2010). Thus, wild animals make decisions often based on environmental cues that lead to movement patterns characteristic of different populations across the landscape (Nathan et al., 2008; Dodge et al., 2014). However, obtaining a comprehensive understanding of large-scale animal movement behavior and population occurrence under climate change scenarios or habitat loss has proven to be a challenge (Araujo & Guisan, 2006). Moreover, while the research toolbox in movement ecology studies has seen a considerable expansion over the last two decades due to technological advancements of the tracking devices and molecular markers (Cushman and Lewis, 2010; Williams et al., 2020), the limitation of scaling up individual data to population-level inferences is still a substantial obstacle (Hawkes, 2009; but see Holdo & Roach, 2013). A promising research approach that may overcome the limitations of wildlife movement studies hindered by small sample sizes is represented by computer simulations within an Agent-based Modeling (ABM) framework (Tang & Bennett, 2010; Bridge et al., 2017).

The core principle of ABM is to simulate a set of entities, called agents, which are defined by intrinsic properties as well as behavioral rules governing their interactions with the environment (Grimm & Railsback, 2013). That is, agents are described by their inherent attributes while dynamically interacting with external conditions such as the co-occurrence of other agents and/or changing features of their environmental setting . Thus, ABM has found applications in many study areas including biology, disease risk, social sciences, and

economics (Polhill et al., 2008; Grimm & Railsback, 2013; Kilmek et al., 2015; Willem et al., 2017) with the unifying goal of investigating and predicting the dynamics of complex systems (Grimm et al., 2005). In particular, wildlife studies have adopted the ABM approach to simulate population growth, reproduction, mortality rate, energy budget, and migration ecology, just to cite a few (Brown & Robinson, 2006; Lustig et al., 2019; Aurbach et al., 2020; Goldstein et al., 2021). However, we currently lack a robust and spatially explicit ABM coding platform for the implementation of large-scale animal movement investigations (but see Thiele et al. (2012) or Chubaty and McIntire (2021)). Here we present a novel ABM framework in the R programming language for applications in animal behavior and movement ecology broadly defined.

2. PACKAGE OVERVIEW

abmR allows for both computation and visualization of agent movement trajectories through a set of behavioral rules based on environmental parameters. The two movement functions, `moveSIM` and `energySIM`, provide the central functionality of the package, allowing the user to run simulations using an Ornstein-Uhlenbeck movement model (Uhlenbeck & Ornstein, 1930; hereafter OU). Additional functions provide a suite of visualization and data summarization tools intended to reduce the effort needed to go from results to presentation-ready figures and tables (Table 1). The package is currently available as a Github repository (https://github.com/bgoch5/abmR), but has been submitted to the Comprehensive R Archive Network (CRAN) to facilitate broader access and usage.

Function	Usage
moveSIM	Runs agent-based model movement simulations based on environmental data
moveVIZ	Creates a plot or table of moveSIM() results
energySIM	Runs agent-based model movement and energy budget simulations based on environmental data
energyVIZ	Creates a plot or table of energySIM() results
tidy_results	Prints results from moveSIM() or energySIM() in an easier-to-read table
get_ex_data	Downloads data that is used in examples in vignette and documentation
as.species	Creates object of class 'species' for input into moveSIM() or energySIM()

Table 1. Functions contained in the abmR package (v. 1.0.1). For more complete function descriptions, consult the abmR manual.

Both movement functions used by abmR rely on the same OU model approach summarized below. Given current agent location (x_t , y_t), agent location at the subsequent timestep (x_{t+1}, y_{t+1}) is modeled according to the following equations:

$$x_{t+1} = x_t + \sigma * Z + \phi_x * (\omega_x - x_t) \quad (1)$$

$$y_{t+1} = y_t + \sigma * Z + \phi_y * (\omega_y - y_t) \quad (2)$$

Here, σ is the randomness parameter from the Brownian motion process that serves as a multiplier on the error term Z, a single random number drawn from the Normal(0,1) distribution. In addition, $\phi_x$and $\phi_y$are movement motivation or attraction strength for the OU process in the longitude and latitude coordinates, respectively, while $\omega_x$and $\omega_y$are optimal x (longitude) and y (latitude) coordinates, respectively. It is assumed that the origin point (x_1 , y_1) is known. The OU model given in (1) and (2) performs similarly to a spring-coil. Greater distance from optimal coordinates $\omega_x$and ω_y acts like a compressed spring to propel distant agents towards $\omega_x$and ω_y. On the other hand, agents closer to $\omega_x$and ω_y will travel a shorter distance on that timestep. However, the amount of movement also depends on $\phi_x$and ϕ_y, because these motivations serve as a multiplier on $(\omega_x - x_t)$ and $(\omega_y - y_t)$, respectively (Eqns. 1 and 2).

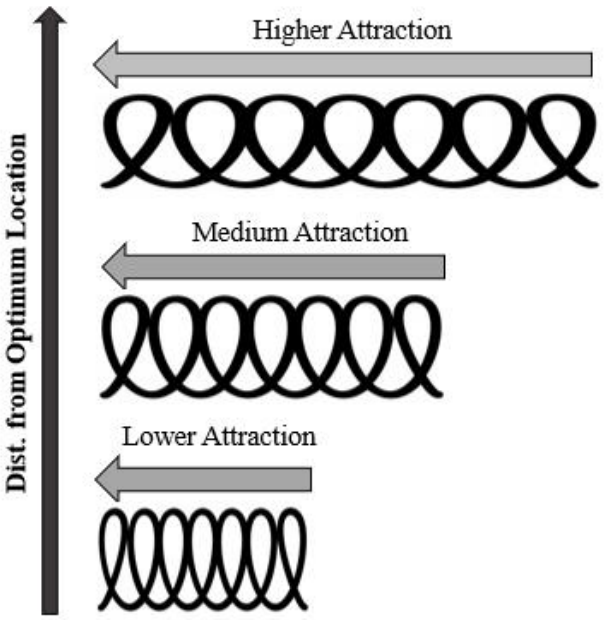


Figure 1. The OU model given in (1) and (2) performs like a spring-coil: agents further from their target location experience higher attraction (and travel further), while agents closer to their destination experience lesser attraction (and travel less far).

Both movement functions allow the user to optionally specify two morphological parameters. These morphological parameters are used to compute adjusted agent motivations ϕ_x and ϕ_y from user-specified motivations ϕ_{x0} and ϕ_{y0}.

$$\phi_x = \phi_{x0} + (\pm 0.1 * ((a - \mu_a)/S_a)) + (\pm 0.1 * ((b - \mu_b)/S_b)) \quad (3)$$

$$\phi_y = \phi_{y0} + (\pm 0.1 * ((a - \mu_a)/S_a)) + (\pm 0.1 * ((b - \mu_b)/S_b)) \quad (4)$$

Here, a, μ_a, and S_a represent the observed value, population mean, and population standard deviation for the first morphological parameter, respectively, while b, μ_b , and S_b represent analogous quantities for the second morphological parameter. The sign on the numerical constant 0.1 depends upon the hypothesized effect of each morphological parameter on agent movement. If increasing values of the parameter lead to increasing motivation, a positive sign

should be used, and if increasing values of the parameter lead to decreasing motivation, a negative sign should be used. For example, if one hypothesizes that increasing mass leads to less movement motivation, a negative sign should be used for this parameter. The biological basis for including morphological parameters as a multiplier on motivation relies on previous research showing that individual and population intrinsic morphological characteristics affect migration speed, distance, and timing in many species (Alerstam, 1993; Hedenström, 2008).

While the two movement functions are distinct (see below), each follows the same basic two steps. The first, large-scale searching, is illustrated in Fig. 2. This step finds the 'optimum' location for each agent ($\omega_x$and $\omega_y$coordinates from Eqns. (1) and (2)), that is, the geographic point with the optimal environmental raster value in a semicircular search region depending on movement orientation. For `moveSIM`, this optimal raster value is supplied directly, while for `energySIM` it is the average of the lower and upper bounds of a user-specified optimum range. The optimum value or range of values specified depends on the modeling scenario and the type of environmental raster that is used (e.g., vegetation, temperature, etc.).

Agents will move toward the selected optimum location. However, if the attraction strength ($\phi_x$and $\phi_y$in Eqns. (1) and (2)) is less than 1, agents will have a 'target' location short of the optimal location. Agents will move towards this target location with some error, which is generated by sampling from the normal distribution and multiplying by σ, as specified by the user (In Fig. 2, σ is 1). Because the support of the normal distribution consists of all real numbers, large deviations from the 'target' point are possible. However, because the normal distribution has low density at the extreme tails, outcomes are most likely to fall within a certain rectangle of the target, as illustrated in Fig. 2. This first step corresponds to the OU model of Eqns. (1) and (2).

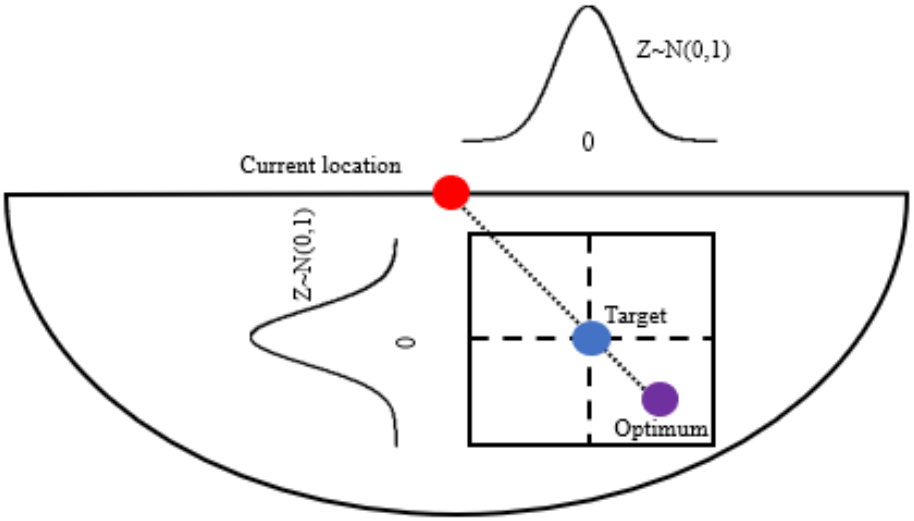


Figure 2. Illustration of large-scale searching specified by the OU model of Eqns. (1) and (2). Agents find an 'optimum' location within the semi-circular search region and then a 'target' location that lies on the line between the 'current location' and the 'optimum' location. If there is a tie between multiple potential 'optimum' cells, one is randomly selected from the list of tied cells to serve as the optimum. Random error is added by sampling from a Z ~ N(0, 1) distribution. Here, $\sigma = 1$ and $\phi_x = \phi_y < 1$, where σ is the multiplier on the random error and ϕ_x and ϕ_y are the motivations in the x and y directions, respectively. Bounding box represents the most probable samples from the N(0,1) distribution.

The second step is small-scale searching. Here, agents select the 'best' of the 8 neighboring cells (queen's case or Moore neighborhood) after performing step 1, discussed above and in Fig 2. Again, 'best' here means the cell with environmental raster value closest to the agent's user-defined optimum range. These two steps are then repeated for each timestep until the agent dies or proceeds through all timesteps. For the `moveSIM` function, agent death occurs when agents fail to achieve suitable environmental raster values for more than a user-specified number of consecutive timesteps. Here, what constitutes a 'suitable' cell is determined by the optimum value and an allowable deviation proportion, both also specified

by the user. For the `energySIM` function, agent death occurs when energy reaches zero. For both functions, users may choose to disable agent mortality. In the following subsections, we present the differences between `moveSIM` and `energySIM` functions and their underlying algorithms.

2.1. Simulation function: moveSIM

The function `moveSIM` runs an OU movement simulation based on environmental conditions provided by the user (e.g., raster), optionally including agent mortality and adjusted motivation according to user-specified morphological parameters. The function operates according to the following algorithm. Here, terms in *italics* are `moveSIM` function arguments (see Table 2).

Argument	Function	Usage
replicates	M, E	number of agents to model
days	M, E	number of timesteps
modeled_species	M, E	Species object from as.species()
env_rast	M, E	Environmental raster
optimum	M	Optimal environmental value
optimum_lo	E	Lowest optimum environmental value
optimum_hi	E	Highest optimum environmental value
dest_x	M, E	Destination Longitude
dest_y	M, E	Destination Latitude
mot_x	M, E	Motivation (x direction)
mot_y	M, E	Motivation (y direction)
search_radius	M, E	Radius of semi-circular search region (km)
direction	M, E	Movement direction: N, S, E, W, or R (Random)
sigma	M, E	Randomness parameter
mortality	M, E	Incorporate agent mortality? T or F
fail_thresh	M	Deviation from optimum constituting failure
n_failures	M	Allowable number of failures before agent death
init_energy	E	Initial energy
energy_adj	E	Energy gain/penalty vector
single_rast	M, E	Using a single-layer raster? T or F
write_results	M, E	Save results as a csv? T or F
x	S	Origin longitude
y	S	Origin latitude
morphpar1	S	Morphological parameter num. 1 observed value
morphpar1mean	S	Morphological parameter num. 1 mean
morphpar1sd	S	Morphological parameter num. 1 SD
morphpar1sign	S	Morphological parameter num. 1 sign
morphpar2	S	Morphological parameter num. 2 observed value
morphpar2mean	S	Morphological parameter num. 2 mean
morphpar2sd	S	Morphological parameter num. 2 SD
morphpar2sign	S	Morphological parameter num. 2 sign

Table 2. List of arguments used in `moveSIM` (M), `energySIM` (E), or `as.species` (S). T: True, F: False, SD: standard deviation. In text, these arguments are presented in *italics*. For a more complete list of argument descriptions, see the abmR documentation.

The following algorithm applies when the argument *direction* is 'N', 'S', 'E', or 'W'. For random movement (*direction* = '*R*') agents simply select a random point from a circle of

radius *search_radius* for each timestep (Step 4). Here, let *env_rast*(x_{t+1},y_{t+1}) be the value of *env_rast* at the point(x_{t+1},y_{t+1}). The core algorithm shown here assumes that *env_rast* contains no undefined (N/A) grid cells.

1. If morphological parameters are specified, compute adjusted motivations $\phi_x$and ϕ_y according to (3) and (4), respectively. If not, set ϕ_x= $\phi_{x0}$and ϕ_y=ϕ_{y0}, where $\phi_{x0}$and $\phi_{y0}$are *mot_x* and *mot_y*, respectively.
2. Specify (x_1, y_1) using *x* and *y* contained in *modeled_species*.
3. Set failures = 0
4. For day t in 1:(*days*-1)
    a. Create a search area defined as a semicircle of radius *search_radius* facing *direction* and centered at (x_t, y_t)
    b. Determine (ω_x,ω_y)as location within the search area with *env_rast* cell value closest to *optimum.* Draw a random sample (n = 1) in case of ties.
        i. If (*dest_x*, *dest_x*) in search area, set (ω_x,ω_y) = (*dest_x*, *dest_y*)
    c. Large scale searching: determine (x_{t+1},y_{t+1})$_0$ according to (1) and (2).
    d. Small scale searching: determine (x_{t+1},y_{t+1}) by selecting location within eight neighboring cells (queen's case) of (x_{t+1},y_{t+1})$_0$ with the cell value closest to *optimum*. Draw a random sample (n = 1) in case of ties.

Perform (e)-(f) if *mortality* = True.

    e. If observed *env_rast*(x_{t+1},y_{t+1}) - *optimum* > *optimum*fail_thresh*, set failures = failures + 1. If not, set failures = 0.

f. If failures > *n_failures*, the agent dies. Set (x_{t+1},y_{t+1}) through (x_{days},y_{days}) as N/A and end loop.

5. Return dataframe with *days* rows and 2 columns movement track data.
6. Repeat (3)-(5) *replicates* times.

2.2. Simulation function: energySIM

The function `energySIM` builds on `moveSIM` by allowing for dynamic agent energy levels that are affected by the quality of environmental values achieved. These initial user-defined energy levels then serve as a driver of mortality and movement distance per timestep. It operates according to the following algorithm. Here, terms in *italics* are `energySIM` function arguments (see Table 2) or calculated variables (e.g., *optimum*, *energy*).

The following algorithm applies when the argument *direction* is 'N', 'S', 'E', or 'W'. For random movement (*direction* = '*R*') agents simply select a random point from a circle of radius *search_radius* for each timestep (Step 5). Here, let *env_rast*(x_{t+1},y_{t+1}) be the value of *env_rast* at the point(x_{t+1},y_{t+1}). The core algorithm shown here assumes that *env_rast* contains no undefined (N/A) grid cells.

1. If morphological parameters are specified, compute adjusted motivations $\phi_x$and ϕ_y according to (3) and (4), respectively. If not, set ϕ_x= $\phi_{x0}$and ϕ_y= ϕ_{y0}, where $\phi_{x0}$and $\phi_{y0}$are *mot_x* and *mot_y*, respectively.
2. Compute *optimum* as (*optimum_hi* - *optimum_lo)/2*.
3. Specify (x_1,y_1) using *x* and *y* contained in *modeled_species*.
4. Set *energy* = *init_energy*
5. For day t in 1:(*days*-1)

a. If *mortality* = True, update *search_radius* as *search_radius* = *search_radius * (energy/init_energy).*

b. Create a search area defined as a semicircle of radius *search_radius* facing *direction* and centered at (x_t, y_t).

c. Determine (ω_x,ω_y)as location within the search area with *env_rast* cell value closest to *optimum.* Draw a random sample (n = 1) in case of ties.

   i. If (*dest_x*, *dest_x*) in search area, set (ω_x,ω_y) = (*dest_x*, *dest_y*)

d. Large scale searching: determine $(x_{t+1},y_{t+1})_0$ according to (1) and (2).

e. Small scale searching: determine (x_{t+1},y_{t+1}) by selecting location within eight neighboring cells (queen's case) of $(x_{t+1},y_{t+1})_0$ with the cell value closest to *optimum*. Draw a random sample (n = 1) in case of ties.

f. If *optimum_lo*<*env_rast*(x_{t+1},y_{t+1})<*optimum_hi* update *energy* = *energy*+*energy_adj*[1].

g. Else, compute Δ = *env_rast*(x_{t+1},y_{t+1}) - *optimum.* And update energy in the following way:

   If Δ< 0.1 * *optimum*, then *energy* = *energy*+*energy_adj*[2]

   If 0.1 * *optimum*< Δ< 0.2 * *optimum*, then *energy* = *energy*+*energy_adj*[3]

   If 0.2 * *optimum*< Δ< 0.3 * *optimum*, then *energy* = *energy*+*energy_adj*[4]

If 0.3 * *optimum*< Δ< 0.4 * *optimum*, then *energy* = *energy*+*energy_adj*[5]

If 0.4 * *optimum*< Δ< 0.5 * *optimum*, then *energy* = *energy*+*energy_adj*[6]

If 0.5 * *optimum*< Δ< 0.6 * *optimum*, then *energy* = *energy*+*energy_adj*[7]

If 0.6 * *optimum*< Δ< 0.7 * *optimum*, then *energy* = *energy*+*energy_adj*[8]

If 0.7 * *optimum*< Δ< 0.8 * *optimum*, then *energy* = *energy*+*energy_adj*[9]

If 0.8 * *optimum*< Δ< 0.9 * *optimum*, then *energy* = *energy*+*energy_adj*[10]

If Δ> 0.9 * *optimum*, then *energy* = *energy*+*energy_adj*[11]

h. If *mortality* = True and *energy* = *0*, the agent dies. Set (x_{t+1},y_{t+1}) through (x_{days},y_{days}) as N/A and end loop.

6. Return dataframe with *days* rows and 2 columns movement track data.
7. Repeat (3)-(6) *replicates* times.

3. EXAMPLE APPLICATIONS

abmR can be used to construct ABM simulations for any desired agent across the globe. In the following example, we demonstrate how `energySIM` can be used to compare the movement and the differential energy allocation of two populations of 250 agents each. While we focus on `energySIM` in this example, a similar workflow applies for `moveSIM`.

3.1. Comparisons between populations

In this example, both populations are characterized by the same number of replicates, movement timesteps ('days'), the same σ, and the same environmental data provided by a Normalized Difference Vegetation Index (NDVI) raster stack containing 14 days of data between September 01-14, 2019 (Vermote, 2019). Both populations had an unspecified destination (indicated with '999'). However, Population 1 (P1) agents started their movements from a different point (105.7° W; 48.2 ° N) situated about 2,800 km from the origin of Population 2 (P2) agents (142.7° W; 63.2 ° N). Additionally, P1 agents had a smaller search radius (150 km) but higher motivation than P2 agents (P1 motivation = 0.95). P1 agents also had different optimum ranges (P1 0.2-0.5; P2 0.6-0.8), and different initial energy units (P1 100; P2 70). These differences in simulation parameterization result in clearly dissimilar movement tracks (Fig. 3).

Population 1

```
am.pop.1 = as.species(x = -105.7, y = 48.2)
sim.move <- energySIM(
  replicates = 250,
  days = 14,
  env_rast = as.raster.stack.ndvi.sep,
  search_radius = 150,
  sigma = 0.1,
  dest_x = 999,
  dest_y = 999,
  mot_x = 0.95,
  mot_y = 0.95,
  modeled_species = am.pop.1,
  optimum_lo = 0.2,
  optimum_hi = 0.5,
  init_energy = 100,
  direction = "S",
  mortality = F,
  Energy_adj = c(20,10,8,5,2,0,-2,-5,-8,-10,-20),
  write_results = T,
  single_rast = F)
```

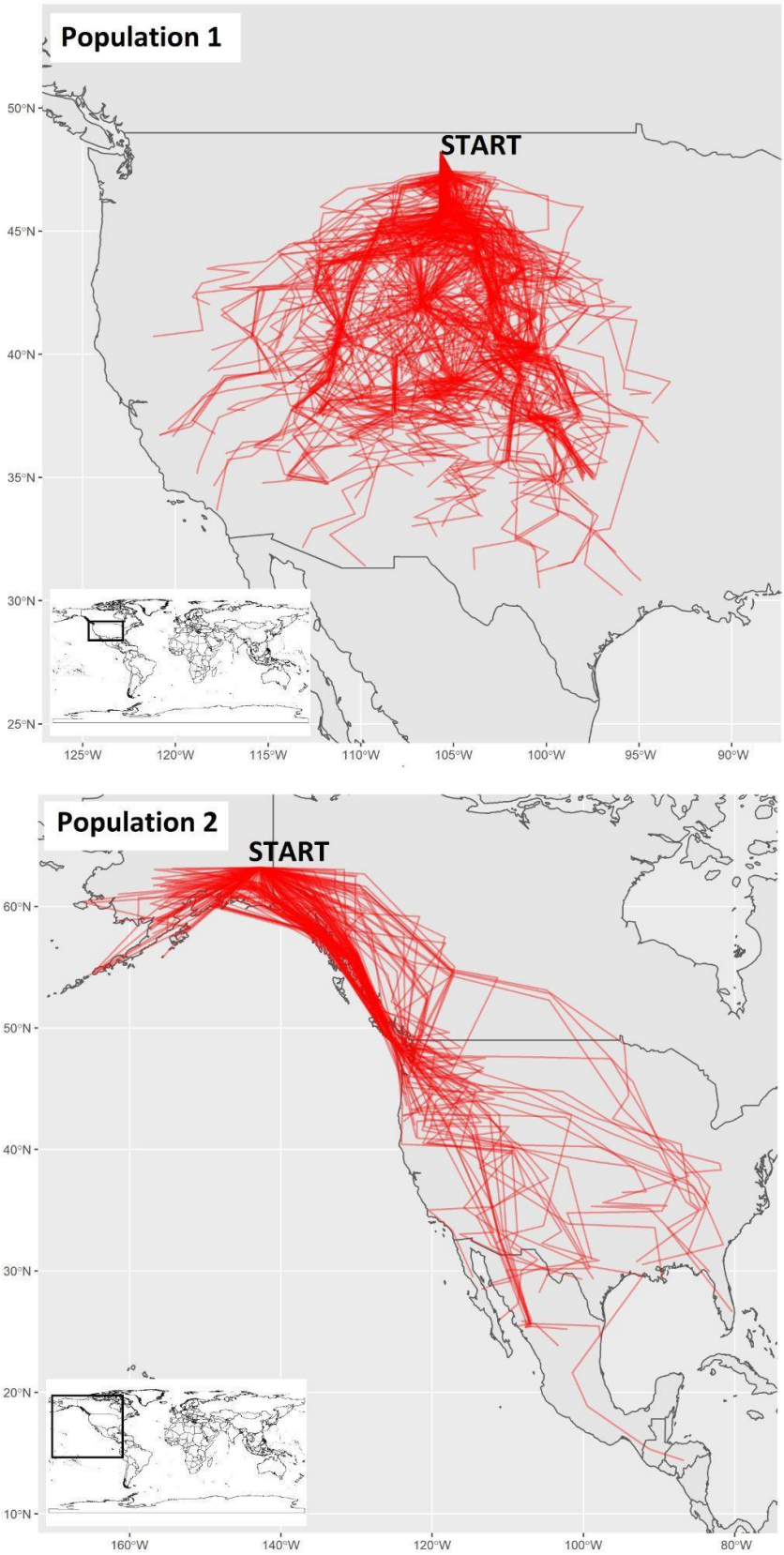
Population 2

```
am.pop.2 = as.species(x = -142.7, y = 63.2)
simtwo.move <- energySIM(
  replicates = 250,
  days = 14,
  env_rast = as.raster.stack.ndvi.sep,
  search_radius = 800,
  sigma = 0.1,
  dest_x = 999,
  dest_y = 999,
  mot_x = 0.8,
  mot_y = 0.8,
  modeled_species = am.pop.2,
  optimum_lo = 0.6,
  optimum_hi = 0.8,
  init_energy = 70,
  direction = "S",
  mortality = F,
  energy_adj = c(20,10,8,5,2,0,-2,-5,-8,-10,-20),
  write_results = T,
  single_rast = F)
```

Box 1. R code used for performing the simulations presented in Fig 3. First, `as.species` is called to initialize two populations with different origin locations (here omitting morphology). Then, `energySIM` is called to perform a movement simulation for each population; parameters that differ

292 between the two simulations are printed in red, functions in blue, and objects in bold. For argument

293 descriptions, see Table 2 and the package manual.



294

Figure 3. Movement tracks reveal that Population 1 tended to travel through the central United States, while Population 2 traveled mostly throughout western Canada, United States, and Mexico. Overall, Population 1 traveled more distance and exhibited more consistent paths near the origin than did Population 2. The movement tracks are natively produced by abmR. Inset world map provided for geographic reference.

While we can compare the movement tracks visually, Table 3 provides a numerical description of results. In this simulation, P1 traveled a much smaller average distance (154.6 km) than did P2 (625.5 km). However, P1 traveled more days on average (7.4 days) before stopping than P2 (3.8 days). Additionally, P2 had higher energy consumption than P1; its average remaining energy across all timesteps was 61.2 units compared to 99.7 units for P1. There are several possible reasons for this observed pattern. First, P2 began with a smaller initial energy (70 units) than P1 (100 units). Additionally, P2 had higher optimum NDVI values (0.6 - 0.8), which might have been less abundant and generally more difficult to reach than those of P1 (0.2 - 0.5). Finally, because they began in different places, P1 and P2 agents encountered different raster cells along their journey.

	variable	mean	sd	median	min	max	range
Population 1	day	7.4	4	7	1	14	13
	longitude	-105.5	4.1	-105.2	-121.1	-93.3	27.8
	latitude	41.6	3.4	41.9	30.2	47.5	17.3
	energy	99.7	1.3	100	80	100	20
	delta energy	-0.04	1.5	0	-20	20	40
	distance	154.6	70	154.4	5.5	316.7	311.1
Population 2	day	3.8	3.08	3	1	14	13
	longitude	-126	17.3	-127.1	-166	-80.3	85.7
	latitude	50.2	10.5	52.2	14.4	63.1	48.7
	energy	61.2	17.2	60	0	100	100
	delta energy	-2.3	9.3	-5	-20	20	40
	distance	625.5	294	649.7	5.5	1445	1439.4

Table 3. A numerical comparison of Populations 1 and 2, created by using values across all timesteps for all agents. 'Day' summarizes the timestep variable of the movement tracks. 'Longitude' and 'latitude' summarize the geographical position of agents, while 'energy' summarizes agents' remaining energy. 'Delta energy' corresponds to the change (gain or loss) of energy between each timestep, while 'distance' refers to the distance traveled between each timestep. This table was produced outside of abmR using raw movement data returned by the package.

Fig. 4 visually compares P1 and P2 movement outputs based on longitude and latitude. This is not a native abmR figure, but rather is produced using the raw data that abmR generates to show the flexible use of the package. In this figure, P1 movements tended to be to the east and south of P2. However, P2 trajectory shows a much wider distribution, with density points extending to the lower values of latitude.

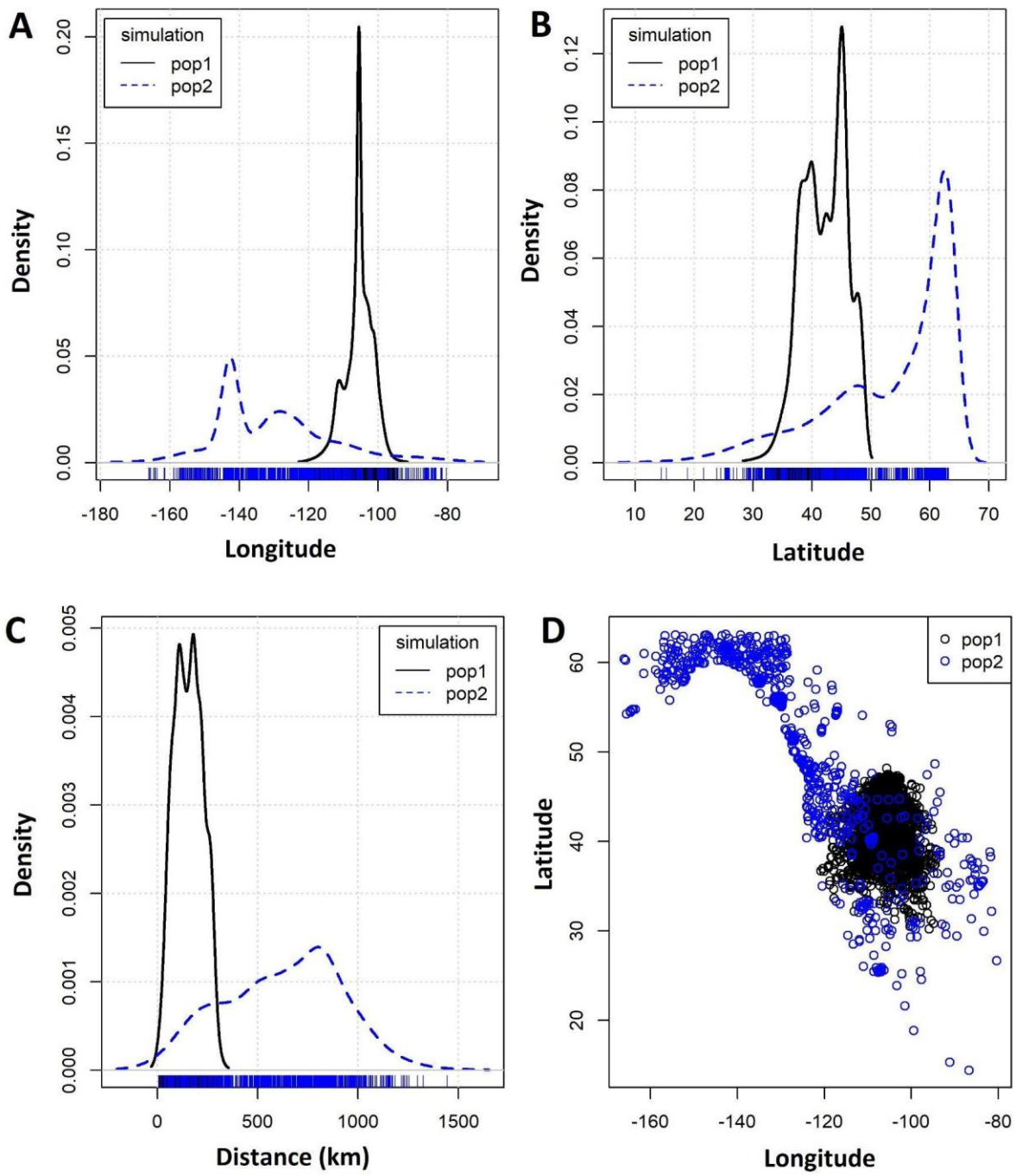


Figure 4. Graphical comparisons of Population 1 and Population 2 movements. Panels A and B show density plots used to individually compare longitude (Panel A) and latitude (Panel B) coordinates attained by agents from each population. Panel C compares the distance traveled between each timestep, while Panel D shows geographical position for all agents in each population across all timesteps.

Finally, Fig. 5 provides a density surface plot for P2 describing agent energy gains (blue) and losses (red) across the landscape. This surface was created using the inverse distance weighted interpolation (IDW) function from the R package 'gstat' (Pebesma, 2004). IDW interpolates grid cell values across a surface using a linear combination of observed (sample) points. When interpolating a cell value, the value of the sample points closer to that cell carry a higher weight, while sample points further from that cell carry smaller weight. IDW is discussed in more detail in Wong (2017). The results from Fig. 5 match well with what we observe in Fig. 3. Movement tracks for P2 tend to follow the blue (energy gain) regions.

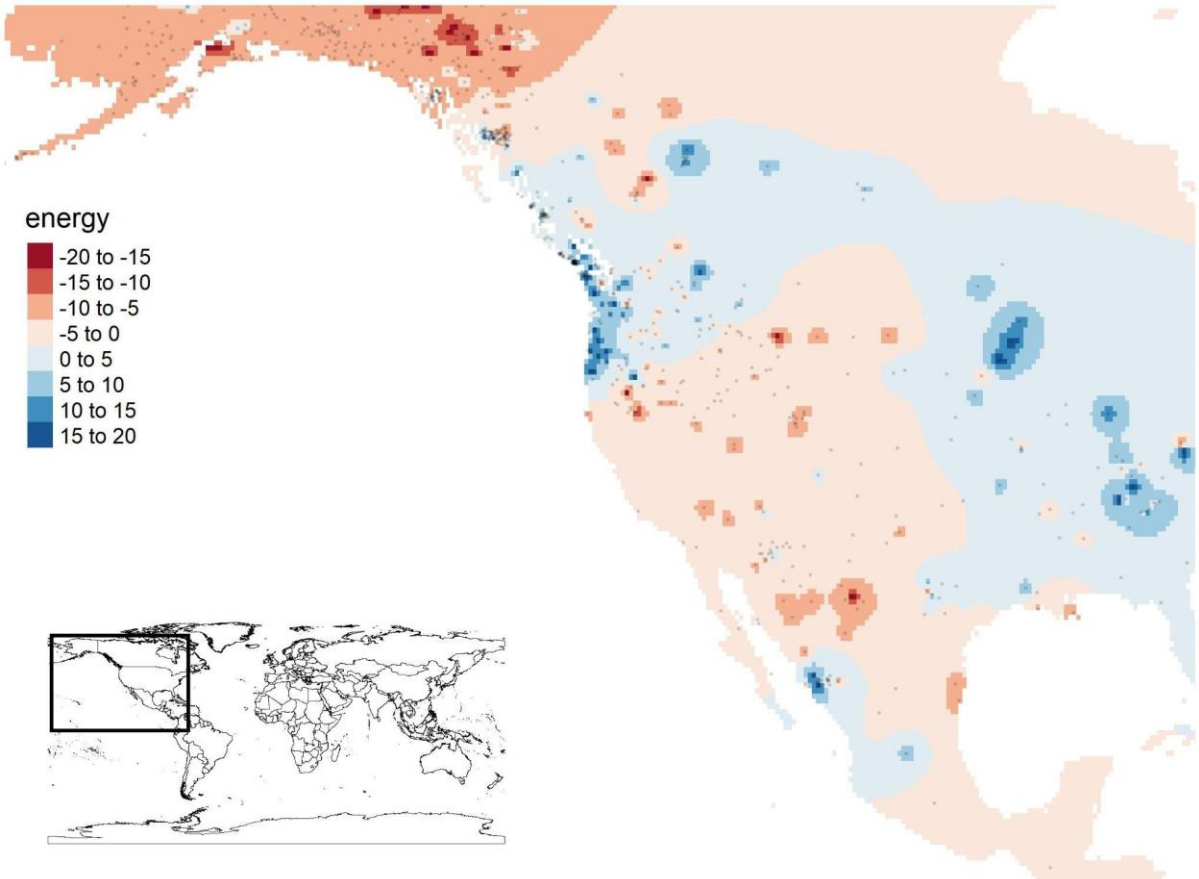


Figure 5. Energy gradient plot of Population 2 by timestep. Areas in red reflect energy loss (less suitable environmental values) while areas in blue reflect energy gain (better environmental values). This plot is produced directly by `energyVIZ`. Inset world map added for geographic reference.

4. CONCLUSIONS AND FUTURE WORK

abmR provides a novel and efficient programming platform for simulating large-scale movements of species across taxa. We ran most of the initial test simulations on a local machine equipped with an Intel® Core™ i7-5500U CPU – 2.40GHz and 8 GB of RAM and obtained results for 100-1000 agents within minutes. The novelty of the software includes the capability of concurrently modeling agent movement trajectories and energy budget. This feature enables a broader exploration of the ecological constraints that shape animal dispersal and/or migration. Moreover, abmR built-in arguments, such as *fail_tresh*, *n_failures*, and *energy_adj*, provide additional flexibility when evaluating mortality scenarios that depend on baseline environmental conditions and energy requirement during prolonged movement bouts (see Table 2 for a full list of arguments affecting mortality).

Over the last decades, spatially explicit simulations and agent-based models have become more popular in ecological and evolutionary studies (Railsback et al., 2006; DeAngelis & Grimm, 2014). Analytical platforms, such as InSTREAM, a simulation model approach designed to understand how stream and river salmonid populations respond to habitat alteration (Railsback et al., 2009), or ALMaSS, a predictive modeling tool for answering environmental policy questions regarding the effect of changing landscape structure on threatened animal species (Topping et al., 2003), allow investigation of specific ecological systems using ABM. On the other hand, many programming languages such as Netlogo, R, or Python are widely used to develop custom and more flexible models that can be adapted to address complex ecological or evolutionary research scenarios (Lustig et al., 2019; Chubaty & McIntire, 2021). However, the use of a programming language to develop a flexible ABM from scratch has two important drawbacks. First, it requires advanced programming skills. Second, its reproducibility can be compromised by the idiosyncrasies of the simulation

algorithm written by the user. These idiosyncrasies, especially if not well documented, can make it difficult or even impossible for other researchers to replicate findings or adapt code to suit their modeling scenarios. abmR provides a novel framework to perform complex movement simulations through standardized functions and arguments that facilitate model annotation and reproducibility while providing publication-ready visualizations at the end of each run.

While we developed and tested abmR as a movement and energy budget simulation tool, its core software functionalities can be adapted to explore other processes such as disease outbreak scenarios (Dougherty et al., 2018). As an example, pathogen vector movement can be easily simulated within abmR, allowing the study of areas of confluence where disease transmission is more probable (Manore et al., 2015). Moreover, potential future updates include the ability to specify multiple raster stacks of different movement predictors and different species traits affecting the movement patterns in the function `as.species`. Additionally, other code expansions might be useful to study plant seed dispersal, interactions of agents (density-dependent scenarios), and altitudinal movements.

Acknowledgements

We thank anonymous reviewers who provided invaluable feedback on an earlier draft. abmR relies heavily on a large set of R packages and we thank the R community for providing support and open-source code.

Author Contributions

B.G. led the development of the package and contributed extensively to the writing of the manuscript; J.F.L. contributed to the package, writing, and critical review of the manuscript;

A.C. conceived the manuscript, led its writing, contributed to the package, and performed package testing.

Conflict of Interest Statement

The authors declare that there are no financial or non-financial conflicts of interest.

Data Availability

abmR and the simulation results for Populations 1 and 2 are available for download from Github at https://github.com/bgoch5/abmR. The environmental raster data used in the examples is available at https://www.ncei.noaa.gov.

Funding

This study was supported in part by research grants from the U.S. National Science Foundation (grants nos. EF 1840230 and DGE 1545261 and DEB 1911955), the National Natural Science Foundation of China (81961128002), and by the Corix Plains Institute.

References

Alerstam, T. (1993). *Bird migration*. Cambridge University Press.

Araujo, M. B. & Guisan, A. (2006). Five (or so) challenges for species distribution modelling. J. Biogeogr. 33: 1677–1688.

Aurbach, A., Schmid, B., Liechti, F., Chokani, N., & Abhari, R. (2020). Simulation of broad front bird migration across Western Europe. Ecological Modelling, 415, 108879.

Berg, H. C. (1983). Random Walks in Biology. Princeton Univ. Press, Princeton, NJ.

Bridge, E.S., Ross, J.D., Contina, A.J., & Kelly, J.F., 2017. Using Agent-Based Models to Scale from Individuals to Populations. In Aeroecology (pp. 259-275). Springer, Cham.

Brown, D. G., & Robinson, D. T. (2006). Effects of heterogeneity in residential preferences on an agent-based model of urban sprawl. Ecology and society, 11(1).

Chubaty A. M., & McIntire E.J.B. (2021). SpaDES: Develop and Run Spatially Explicit Discrete Event Simulation Models. R package version 2.0.7. https://CRAN.R-project.org/package=SpaDES

Cushman, S. A., & Lewis, J. S. (2010). Movement behavior explains genetic differentiation in American black bears. Landscape ecology, 25(10), 1613-1625.

DeAngelis, D. L., & Grimm, V. (2014). Individual-based models in ecology after four decades. F1000prime reports, 6.

Dingle, H. (2014). Migration: the biology of life on the move. Oxford University Press, USA.

Dodge, S., Bohrer, G., Bildstein, K., Davidson, S.C., Weinzierl, R., Bechard, M.J., Barber, D., Kays, R., Brandes, D., Han, J., & Wikelski, M. (2014). Environmental drivers of variability in the movement ecology of turkey vultures (*Cathartes aura*) in North and South America. Philosophical Transactions of the Royal Society B: Biological Sciences, 369(1643), p.20130195.

Dougherty, E.R., Seidel, D.P., Carlson, C.J., Spiegel, O., & Getz, W.M., 2018. Going through the motions: incorporating movement analyses into disease research. Ecology letters, 21(4), pp.588-604.

Giuggioli, L., & Bartumeus, F. (2010). Animal movement, search strategies and behavioural ecology: a cross-disciplinary way forward. Journal of Animal Ecology, 79(4), 906-909.

Goldstein, E., Erinjery, J. J., Martin, G., Kasturiratne, A., Ediriweera, D. S., de Silva, H. J., ... & Iwamura, T. (2021). Integrating human behavior and snake ecology with agent-based models to predict snakebite in high risk landscapes. PLoS neglected tropical diseases, 15(1), e0009047.

441 Grimm, V., Revilla, E., Berger, U., Jeltsch, F., Mooij, W. M., Railsback, S. F., ... &
442 DeAngelis, D. L. (2005). Pattern-oriented modeling of agent-based complex systems: lessons
443 from ecology. science, 310(5750), 987-991.

444 Grimm, V., & Railsback, S. F. (2013). Individual-based modeling and ecology. Princeton
445 university press.

446 Hawkes, C. (2009). Linking movement behaviour, dispersal and population processes: is
447 individual variation a key?. Journal of Animal Ecology, 78(5), 894-906.

448 Hedenström, A. (2008). Adaptations to migration in birds: behavioural strategies,
449 morphology and scaling effects. Philosophical Transactions of the Royal Society B:
450 Biological Sciences, 363(1490), pp.287-299.

451 Holdo, R. M., & Roach, R. R. (2013). Inferring animal population distributions from
452 individual tracking data: theoretical insights and potential pitfalls. Journal of Animal
453 Ecology, 82(1), 175-181.

454 Klimek, P., Poledna, S., Farmer, J. D., & Thurner, S. (2015). To bail-out or to bail-in?
455 Answers from an agent-based model. Journal of Economic Dynamics and Control, 50, 144-
456 154.

457 Lustig, A., James, A., Anderson, D., & Plank, M. (2019). Pest control at a regional scale:
458 Identifying key criteria using a spatially explicit, agent-based model. Journal of Applied
459 Ecology, 56(7), 1515-1527.

460 Manore, C.A., Hickmann, K.S., Hyman, J.M., Foppa, I.M., Davis, J.K., Wesson, D.M., &
461 Mores, C.N. (2015). A network-patch methodology for adapting agent-based models for
462 directly transmitted disease to mosquito-borne disease. Journal of biological dynamics, 9(1),
463 pp.52-72.

464 Nathan, R., Getz, W. M., Revilla, E., Holyoak, M., Kadmon, R., Saltz, D., & Smouse, P. E.
465 (2008). A movement ecology paradigm for unifying organismal movement research.
466 Proceedings of the National Academy of Sciences, 105(49), 19052-19059.

467 Pebesma, E.J. (2004). Multivariable geostatistics in S: the gstat package. Computers &
468 Geosciences, 30: 683-691.

469 Polhill, J. G., Parker, D., Brown, D., & Grimm, V. (2008). Using the ODD protocol for
470 describing three agent-based social simulation models of land-use change. Journal of
471 Artificial Societies and Social Simulation, 11(2), 3.

472 Railsback, S. F., Lytinen, S. L., & Jackson, S. K. (2006). Agent-based simulation platforms:
473 Review and development recommendations. Simulation, 82(9), 609-623.

474 Railsback, S. F., Harvey, B. C., Jackson, S. K., & Lamberson, R. H. (2009). InSTREAM: the
475 individual-based stream trout research and environmental assessment model. Gen. Tech. Rep.

476 PSW-GTR-218. Albany, CA: US Department of Agriculture, Forest Service, Pacific
477 Southwest Research Station. 254 p, 218.

478 Tang, W., & Bennett, D.A. (2010). Agent-based modeling of animal movement: a review.
479 Geography Compass, 4(7), pp.682-700.

480 Thiele, J. C., Kurth, W., & Grimm, V. (2012). RNetLogo: An R package for running and
481 exploring individual-based models implemented in NetLogo. Methods in Ecology and
482 Evolution, 3(3), 480-483.

483 Topping, C. J., Hansen, T. S., Jensen, T. S., Jepsen, J. U., Nikolajsen, F., & Odderskær, P.
484 (2003). ALMaSS, an agent-based model for animals in temperate European landscapes.
485 Ecological Modelling, 167(1-2), 65-82.

486 Uhlenbeck, G. E., & Ornstein, L. S. (1930). On the theory of the Brownian motion. Physical
487 review, 36(5), 823

488 Vermote, Eric; NOAA CDR Program. (2019): NOAA Climate Data Record (CDR) of
489 AVHRR Normalized Difference Vegetation Index (NDVI), Version 5. NOAA National
490 Centers for Environmental Information. https://doi.org/10.7289/V5ZG6QH9.

491 Willem, L., Verelst, F., Bilcke, J., Hens, N., & Beutels, P. (2017). Lessons from a decade of
492 individual-based models for infectious disease transmission: a systematic review (2006-
493 2015). BMC infectious diseases, 17(1), 1-16.

494 Williams, H. J., Taylor, L. A., Benhamou, S., Bijleveld, A. I., Clay, T. A., de Grissac, S., ...
495 & Börger, L. (2020). Optimizing the use of biologgers for movement ecology research.
496 Journal of Animal Ecology, 89(1), 186-206.

497 Wong, D. W. (2016). Interpolation: Inverse-Distance Weighting. International Encyclopedia
498 of Geography: People, the Earth, Environment and Technology: People, the Earth,
499 Environment and Technology, 1-7.

500

501

502

503

504